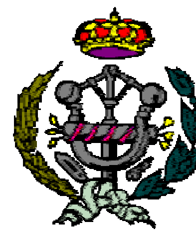


**UNIVERSIDAD POLITÉCNICA DE VALENCIA**

**ESCUELA UNIVERSITARIA DE INGENIERIA TÉCNICA INDUSTRIAL**

# MANUAL DE USUARIO DEL MÓDULO MICROCONTROLADOR DALLAS 5000(T).



# ÍNDICE DE PÁGINAS.

# ÍNDICE DE FIGURAS.

# ÍNDICE DE TABLAS.

# 1. INTRODUCCIÓN.

Vamos a explicar de manera puntual los pasos a seguir para la transmisión de programas desde el PC al microcontrolador. Para ello vamos a necesitar dos de los módulos diseñados, estos son:

El módulo de comunicaciones.

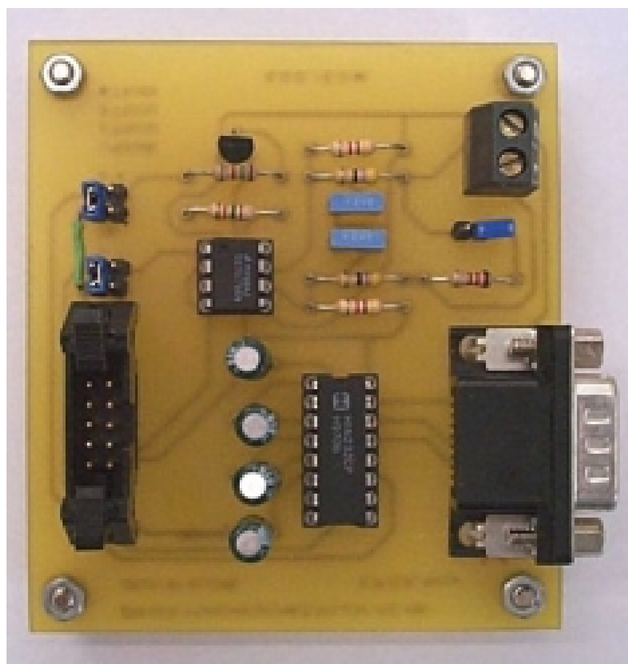


**Figura 1. Vista del módulo de comunicaciones.**

El módulo microcontrolador.

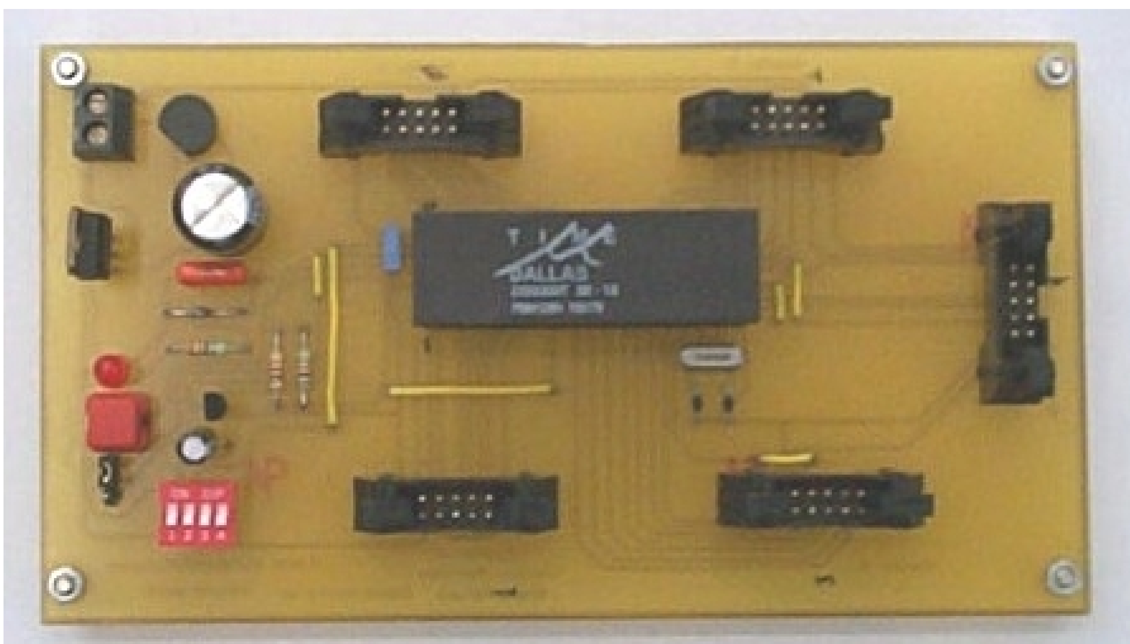


**Figura 2. Vista del módulo microcontrolador.**

Suponemos que los dos módulos están montados siguiendo los esquemáticos y los fotolitos proporcionados, y que además de los módulos anteriores, poseemos un cable serie que conecte el módulo de comunicaciones con el PC, y tenemos una cinta Bus que nos conectaran ambos módulos.

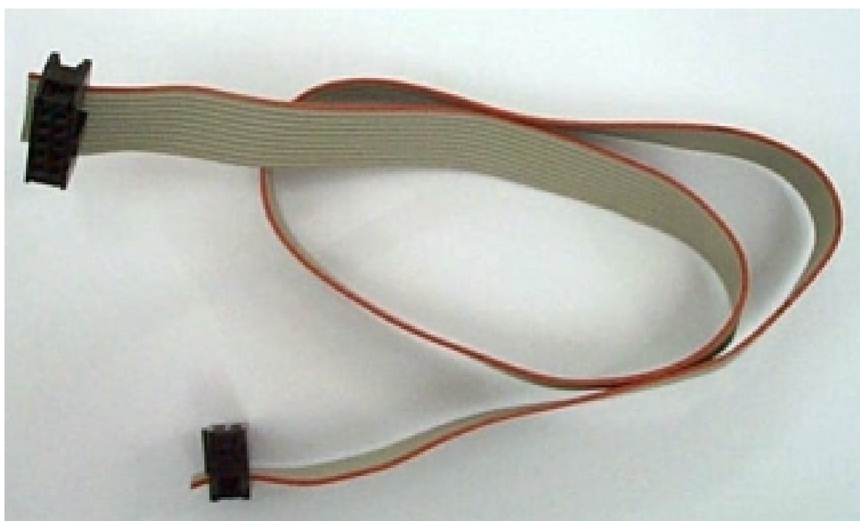


**Figura 3. Fotografía del cable Cinta Bus utilizado**

Se recuerda que la configuración de este cable es la siguiente:

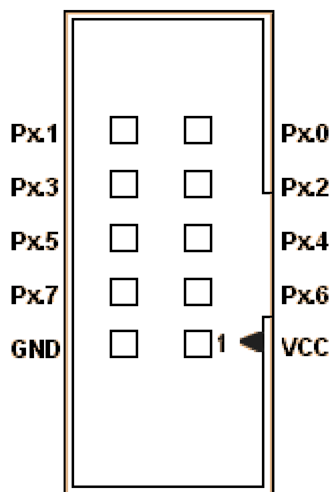



**Figura 4. Señales del conector.**

# 2. DESCRIPCIÓN.

El primer paso a seguir será la conexión del cable serie desde el ordenador al módulo de comunicaciones, para ello, y como medida de protección conectaremos uno de los extremos del cable serie al PC, estando este APAGADO, pues podríamos causar cortocircuitos en el PC, que podrían dar al traste con los puertos serie del mismo. Una vez solucionado este tema, conectaremos el otro extremo del cable serie al módulo de comunicaciones. Interconectaremos los dos módulos mediante un cable cinta BUS, conectándolo al puerto 3 de la placa microcontroladora.

Una vez realizado tal conexión, vamos a terminar de configurar la placa, para ello, y dependiendo el microcontrolador elegido, deberemos seleccionar el tipo de reset adecuado para el sistema, para ello moveremos el jumper de la siguiente manera:

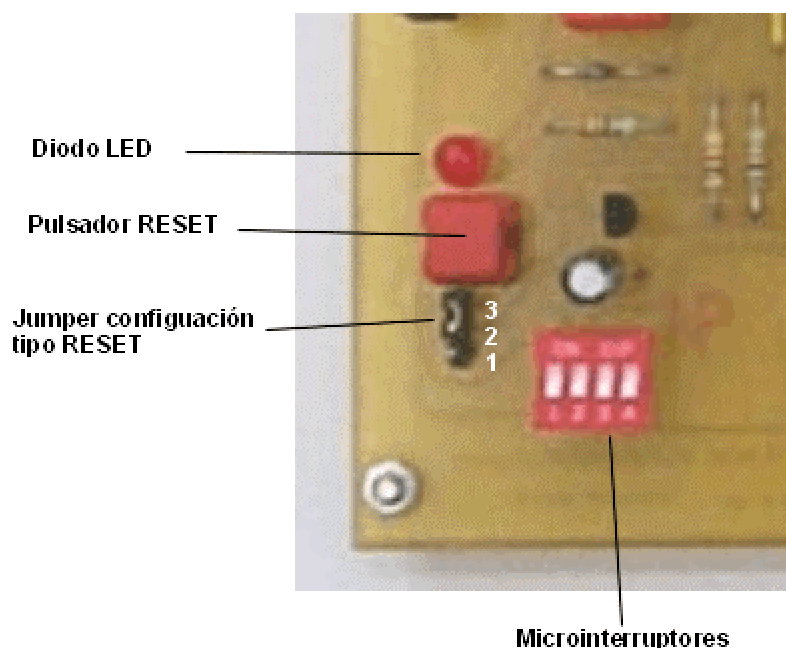



**Figura 5.Posicionamiento de los elementos de configuración.**

| Jumper | |
|---|---|
| **1-2** | Reset a través de un condensador |
| **2-3** | Reset a través de una resistencia |

**Tabla 1. Configuración del jumper selector del Reset.**

Además de esta pequeña configuración, queda otra todavía más importante si cabe, es la que va a determinar el funcionamiento del microcontrolador. Se trata del valor que tiene que tener el pin $\overline{EA}$, para ello, después de mirar en las hojas de características de nuestro micro, procederemos a la configuración manual de los microinterruptores que incorpora la placa. Su funcionamiento se recoge en la siguiente tabla:

| Microinterruptor | Función |
|---|---|
| **1** | ON: Fuerza un '1' en el reset (modo programación en el Dallas 5000). |
| | OFF: Modo normal de funcionamiento. |
| **2** | ON: $\overline{EA}$ = 1, modo de funcionamiento del Dallas 5000. |
| | OFF: $\overline{EA}$ = 0, |
| **3** | Libre para futuras ampliaciones. |
| **4** | Libre para futuras ampliaciones. |

**Tabla 2. Configuración de los microinterruptores.**

Ahora ya podremos conectar el PC y configurar el programa de comunicaciones adecuado. Como en el 90 % de todos los ordenadores actuales, el sistema operativo por excelencia es la plataforma Windows en todas sus versiones, este sistema operativo lleva incluido una utilidad que se llama HyperTerminal, el cual puede leer los puertos serie del PC.

Los valores con los cuales configuraremos este tipo de programas son:

- Bits por segundo: 9600, si a esta velocidad se produjeran problemas, bajaremos este valor.
- Bits de datos: 8.
- Paridad: Ninguna.
- Bits de parada: 1.
- Control de flujo: Ninguno.

Con todos estos valores ya configurados y puesto en modo programación, nos debería aparecer una pantalla similar a esta, al darle al intro en el PC:

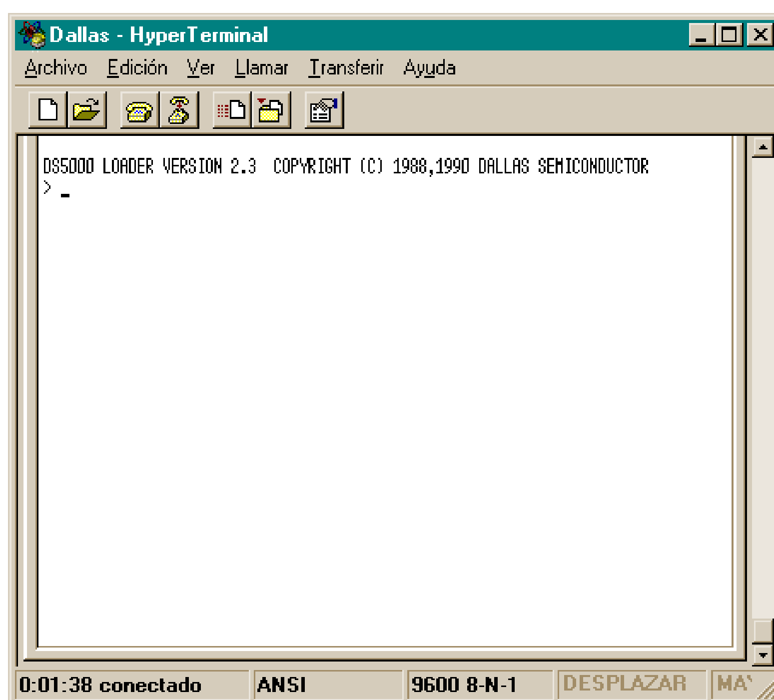



**Figura 6. Pantalla del Hyperterminal de Windows.**

A partir de esta pantalla ya podemos empezar a trabajar con el microcontrolador.

Como se puede observar en la imagen, el microcontrolador nos ha devuelto una línea característica del programa invocado para la transmisión, ya sea esta serie o paralela. En concreto, aquí se ha utilizado un microcontrolador de la casa Dallas, el modelo Dallas 5000. Cuando el microcontrolador se encuentre listo para recibir el próximo comando, nos lo hará saber a través de este símbolo ">".

## 3. COMANDOS.

Este tipo de micro implementa una interfaz en línea de comandos de fácil uso y que es similar a la utilizada por numerosos programas de este estilo. En concreto los comandos que utiliza este modelo son:

| Comando | Función |
|---|---|
| **C** | Retorno de CRC-16 chequeo de la RAM interna. |
| **D** | Visualizar la RAM en hexadecimal del formato de INTEL. |
| **F** | Rellenar la RAM interna con un valor constante. |
| **G** | Obtener el valor actual de un puerto |
| **K** | Cargar 40-bits de código encriptado. |
| **L** | Cargar fichero hexadecimal en formato INTEL. |
| **P** | Poner un valor en un puerto. |
| **R** | Leer el registro MCON. |
| **T** | Refleja datos hexadecimales en formato INTEL. |
| **U** | Borrar la clave de seguridad. |
| **V** | Verificar la RAM interna en hexadecimal. |
| **W** | Escribir en el registro MCON. |
| **Z** | Poner la clave de seguridad |
| **^C** | Interrupción de cualquier operación que se este realizando. |
| **Xon/Xoff** | Control del flujo de la transmisión. |

**Tabla 3. Tabla resumen de los comandos del DS5000(T).**

Pasamos brevemente a dar una explicación más detallada de estos comandos:

**C:** Sintaxis: C[Dirección Inicial [Dirección Final]]
Este comando nos va a devolver un valor que va a ser el CRC de la RAM que le se le indique.

**D:** Sintaxis: D[Dirección Inicial [Dirección Final]]
Visualiza el rango de memoria indicado ( es opcional), en formato hexadecimal de INTEL.

**F:** Sintaxis: F byte [Dirección Inicial [Dirección Final]]
Rellena la memoria con un valor en bytes el rango seleccionado, que es opcional.

**G:** Sintaxis: G
Lectura de los datos que contienen los puertos 0, 1, 2 y 3 y son mostrados en grupos de dos pares de valores hexadecimales.

**K:** Sintaxis: K byte-1 byte-2 byte-3 byte-4 byte-5
Carga la calve de encriptación. Los cinco bytes son mostrados antes de que sean puestos en los registros.

**L:** Sintaxis: L
Carga de ficheros .hex, con el formato INTEL. Cada byte escrito en memoria es verificado a posteriori, si ocurriese que se encontraran valores diferentes se produciría de inmediato un error.

**P:** Sintaxis: P <Valor de P0> <Valor de P1> <Valor de P2> <Valor de P3>
Escribe los valores introducidos en todos los puertos simultáneamente.

**R:** Sintaxis: R
Muestra el valor del registro MCON.

**T:** Sintaxis: T
Sigue la entrada de ficheros con formato .hex de INTEL. Su valor inicial es OFF.

**U:** Sintaxis: U
Limpia el "Security Lock" o clave de seguridad. La variable RANGE es puesta a 32K y la partición es puesta para toda la memoria de programas. Nota: mientras de esta desbloqueando el "Security Lock", se ponen a cero el registro de encriptación, el Vector RAM y el bit $\overline{\text{CE1}}$de la RAM. Los comandos U y Z son los únicos comandos que pueden ser ejecutados cuando el chip esta bloqueado.

**V:** Sintaxis: V

Verifica el contenido de la memoria RAM con el formato hexadecimal INTEL. Este comando opera de manera similar al comando L (load), excepto que este no escribe, simplemente compara byte de la memoria con byte de la fuente.

**W:** Sintaxis: W byte

Escribe un byte en el registro MCON, para configurar la Partición, el Rango y el bit ECE2. Los bits PAA y SL no son afectados por este comando.

**Z:** Sintaxis: Z

Pone a '1' el "Security Lock". Solo los comandos U y Z pueden ser ejecutados después.

**^C:** Sintaxis: ^C

Hace un reset de todas las operaciones.

**Xon/Xoff:** Sintaxis: Xon/Xoff

Estos dos comandos ofrecen el control del flujo de la transmisión. En el modo de transmisión serie, no es necesario ninguno de ellos, aunque puede responder a los dos.

Una vez ya conocemos el lenguaje que empleará nuestro microcontrolador, vamos a explicar como introducirle un programa, que antes de todo, deberá haber sido compilado y enlazado, obteniendo el fichero .hex.